

Amendment to Claims

1. (Currently Amended) A method of processing electronic commerce transactions comprising messages exchanged between a client and a server of a computer network, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a request message from the client at the eTA system, the request message relating to an aspect of the electronic commerce transaction;

extracting data from the request message to record a state of the electronic commerce transaction;

detecting that a failure has occurred with respect to the electronic commerce transaction;

determining whether an outcome of the electronic commerce transaction in relation to the request message has failed, and the actual state of the electronic commerce transaction at the failure;

selecting an appropriate recovery action to recover from the failure based upon said actual state;

transmitting a response message to the client in accordance with the recovery action, wherein the response message masks the failure from the client by providing an expected response to the request message from the client.

2. (Currently Amended) A method as defined in claim 1, additionally comprising identifying a transaction type associated with the electronic commerce transaction.

3. (Original) A method as defined in claim 1, wherein each transaction type has an associated transaction model, and additionally comprising maintaining a data base of transaction models and identifying a transaction type by selecting from a transaction model in the database.

4. (Original) A method as defined in claim 3, wherein a transaction model defines an expected response message from the server for a given request message from the client to thereby enable detecting that a failure has occurred with respect to the transaction when the expected response message is not received.

5. (Original) A method as defined in claim 3, wherein a transaction model defines suspicious activity such that fraudulent activity is deemed present when the suspicious activity is encountered in a transaction.

6. (Original) A method as defined in claim 1, wherein it is deemed that a failure has occurred with respect to the transaction when a response message is not received from the server in response to the request message.

7. (Original) A method as defined in claim 1, wherein a failure has occurred with respect to the transaction when an error code is contained within a response message from the server.

8. (Original) A method as defined in claim 1, additionally comprising discarding data that relates to a transaction state that is stored at the server.

9. (Original) A method as defined in claim 1, wherein determining whether an outcome of the transaction in relation to the request message has succeeded or failed comprises sending a query message to the server to inquire as to the state of the transaction.

10. (Currently Amended) A method as defined in claim 1, wherein the appropriate recovery action comprises re-directing the ~~transaction~~ request message to another server in order to complete the failed transaction.

11. (Currently Amended) A method as defined in claim 1, wherein the commerce electronic transaction relates to ~~a commerce transaction comprised of~~ adding an item to a shopping cart, and wherein extracting data from the request message to record a state of the electronic transaction comprises recording the contents of the shopping cart using data contained in the request message.

12. (Original) A method as defined in claim 1, wherein the response message masks the failure from the client such that the client is oblivious to the failure.

13. (Original) A method as defined in claim 1, wherein the response message masks the failure from the client such that the client is compensated for the failure.

14. (Currently Amended) A method of processing electronic commerce transactions comprising messages exchanged between a client and a server of a computer network, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system and initiating a series of processes at the eTA system, the processes including:

a transaction monitoring process wherein the eTA system monitors electronic commerce messages that are exchanged between the client and the server in relation to a transaction;

a state capture process wherein the eTA system captures and records information descriptive of one or more states of the transaction;

a failure detection process wherein the eTA system determines that a failure has occurred with respect to the transaction and the actual state of the transaction at failure;

an outcome determination process wherein the eTA system determines the extent to which the server has processed the transaction;

a failure masking process wherein the eTA system masks the occurrence of the failure from the client by sending a response message to the client that is an expected response that the client would have received had the failure not occurred; and

a transaction recovery process wherein the eTA system recovers the transaction from the failure based upon said actual state.

15. (Original) A method as defined in claim 14, wherein the state capture process comprises capturing packets contained in electronic request messages from the client to the server and storing the packets with an identifier associated with a particular transaction between the client and the server.

16. (Original) A method as defined in claim 14, wherein the failure detection process comprises monitoring for a failure code that is embedded in a response message from the server, wherein the failure code indicates that a failure has occurred.

17. (Original) A method as defined in claim 14, wherein the failure detection process comprises monitoring for a response message from the server and deeming that a failure has occurred if a response message is not received within a predetermined time span.

18. (Cancelled)

19. (Currently Amended) A method of processing network messages between a network client and a network server, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a network message at the eTA system, which is responsible for the communications between the network client and the network server;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction;

indicating a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period;

determining the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state to recover from the detected failure;

providing a response message to the network client with an appropriate message corresponding to the correct outcome to mask the detected failure; and

logging and reporting relevant information about the state and the message parameters of the electronic commerce transaction.

20. (Original) A method as defined in claim 19, wherein the communications connection is a secure connection.

21. (Original) A method as defined in claim 19, wherein indicating a detected failure comprises monitoring operation of hardware and software components of the communication connection.

22. (Original) A method as defined in claim 21, wherein monitoring comprises intercepting responses from the back-end servers and inspecting the enclosed messages to check for failures and formulating an appropriate response and sending it to the network client.

23. (Original) A method as defined in claim 19, wherein the network messages are transmitted in accordance with Internet protocol processing.

24. (Original) A method as defined in claim 19, further including training the transaction assurance system to classify and identify transaction types using a supervised machine learning technique, thereby enabling the system to be deployed in different e-business environments with different transaction models.

25. (Currently Amended) A method as defined in claim 24, wherein a transaction model is associated with a type of electronic commerce transaction such that the transaction model defines expected network activity with respect to the associated type of electronic commerce transaction.

26. (Original) A method as defined in claim 25, wherein the expected network activity comprises response messages that are expected from the server in response to request messages from the client.

27. (Currently Amended) A method as defined in claim 25, additionally comprising detecting ~~for~~ a failure in a network backend system by comparing a response message from the backend system to an expected response message defined in a transaction model.

28. (Currently Amended) A method as defined in claim 25, wherein determining ~~an expected~~ the correct outcome of the transaction is accomplished by determining an expected outcome that is defined in a transaction model.

29. (Currently Amended) A method as defined in claim 25, providing a response message to the network client with an appropriate message corresponding

to the expected outcome to mask the detected failure is accomplished by using a response message that is defined in a transaction model.

30. (Original) A method as defined in claim 25, wherein a transaction model defines suspicious activity and additionally comprising determining that fraudulent activity is present when the suspicious activity is encountered in a transaction.

31. (Original) A method as defined in claim 25, wherein a transaction model defines a billing charge for a type of transaction and additionally comprising tabulating billing charges based on the number of times that an actual transaction defined in a transaction model is encountered.

32. (Original) A method as defined in claim 19, wherein the system permits resumption of communication with wireless clients when the wireless clients reconnect to the system, without having to resubmit requests they made before disconnecting from the system due to losing wireless signal.

33. (Currently Amended) An electronic transaction assurance (eTA) system that includes:

a communications processor that receives electronic commerce transaction messages over a computer network between a customer at a client node and a server node; and

a policy-based policy manager engine that manages electronic commerce transaction message processing and resulting customer experience by allowing users of the system to define message processing policies that specify conditions and actions to be taken when any of the specified policy conditions is true to provide transparent failover by masking failures from the customer, said masking comprising providing a response message to the customer in accordance with said policies.

34. (Original) An electronic transaction assurance system as defined in claim 33, wherein the policy manager engine masks computer network failures from the customer and generates message interaction with the customer if needed to keep the customer informed of any processing delays and keep the customer engaged in a message dialog to enhance the customer's interaction experience with an e-business Web site at the server node.

35. (Original) A system as defined in claim 34, wherein the eTA system includes multiple eTA nodes, each including a communications processor and a policy manager engine.

36. (Currently Amended) A system as defined in claim 35, wherein at least one of the policy manager engines includes a transaction model that is formulated and built to enable tracking processing of an electronic commerce transaction and storing transaction state, and sharing this state with other eTA nodes in the system.

37. (Currently Amended) A method for determining the outcome of an electronic commerce transaction initiated by a network message between a network client and a network server, the method comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a network message related to said electronic commerce transaction at the eTA system, which is responsible for the communications between the network client and the network server;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

generating a transaction identifier associated with ~~each electronic transaction~~ the received message from the client and storing the transaction identifier information with the transaction type and message parameters at a back end database;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction; and

resuming the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the received message.

38. (Original) A method as defined in claim 37, wherein the transaction identifier is stored in an existing field of the back end database.

39. (Original) A method as defined in claim 37, wherein the transaction identifier is stored in a database table of the back end database.

40. (Original) A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using an applet executing at the network client.

41. (Original) A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using an Internet cookie.

42. (Original) A method as defined in claim 37, wherein storing the transaction identifier comprises inserting information into the back end server database using a browser program at the network client.

43. (Currently Amended) A method for measuring the end-to-end response time of each electronic transaction message sent from a network client side to a network server side of a Web site, the method comprising:

establishing a communications connection between the network client and the Web site network server;

receiving a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message parameters, thereby defining an electronic commerce transaction to which the message relates;

adding code to the Web page served to the network client that records the time when a client request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response message is received by the network client, indicating the end of a said electronic commerce transaction;

generating a transaction identifier associated with each electronic commerce transaction request message received from the network client and storing the transaction identifier information with the transaction type and message parameters at a back end database;

preserving a state of the electronic transaction and updating the transaction type and message parameters in response to processing of the electronic transaction; and

resuming the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the request message from the network client.

44. (Original) A method as defined in claim 43, wherein the transaction identifier is stored in a database table of the back end database.

45. (Original) A method as defined in claim 43, wherein storing the transaction identifier comprises inserting information into the back end server database using an applet executing at the network client.

46. (Original) A method as defined in claim 43, wherein storing the transaction identifier comprises inserting information into the back end server database using an Internet cookie.

47. (Original) A method as defined in claim 43, wherein storing the transaction identifier comprises inserting information into the back end server database using a browser program at the network client.

48. (Original) A method as defined in claim 43, wherein the eTA system includes multiple eTA nodes.

49. (Original) A method as defined in claim 48, wherein a received network message is directed to one of the available eTA nodes.

50. (Original) A method as defined in claim 43, further including:

- communicating information relating to the communications connection at the selected node to one or more of the remaining eTA nodes;
- detecting the removal or failure of the selected eTA node from operation during processing of the received network message;
- preserving the state of the selected eTA node processing with respect to the received network message in one or more nodes of the eTA system; and
- moving the communications connection from the removed selected node to one of the remaining eTA nodes that are still operating in accordance with the preserved node state such that the network client and network server that were using the selected eTA node do not see any interruption in their communications.

51. (Currently Amended) A program product for use in a processor that executes program steps recorded in a computer-readable media to perform a method of processing network messages between a network client and a network server, the program product comprising:

- a recordable media;
- a program of computer-readable instructions executable by the processor to perform operations comprising:
 - establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a network message at the eTA system, which is responsible for the communications between the network client and the network server;

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction;

indicating a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period;

determining the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state and selecting an appropriate action to recover from the detected failure;

providing a an expected response message to the network client ~~with an appropriate message~~ to mask the detected failure;

logging and reporting relevant information about the state and the message parameters of the electronic commerce transaction.

52. (Currently Amended) A system that processes network messages between a network client and a network server, the system comprising one or more

processors that execute program instructions and receive a data set, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receive a network message at the eTA system, which is responsible for the communications between the network client and the network server;

identify a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

preserve a state of the electronic commerce transaction and update the transaction type and message parameters in response to processing of the electronic commerce transaction;

indicate a detected failure in a network back-end system or the network communications connection in response to inspection of the content of a received response from back-end system servers or the lack of a received response within a predetermined time period;

determine the correct outcome of the electronic commerce transaction as affected by the detected failure and the state of the electronic commerce transaction at the failure, and selecting an appropriate action based upon said state and selecting an appropriate action to recover from the detected failure;

provide a an expected response message to the network client with an appropriate message to mask the detected failure; and

log and report relevant information about the state and the message parameters of the electronic commerce transaction.

53. (Currently Amended) A program product for use in a processor that executes program steps recorded in a computer-readable media to perform a method for determining the outcome of an electronic commerce transaction initiated by a network message between a network client and a network server, the program product comprising:

a recordable media;

a program of computer-readable instructions executable by the processor to perform operations comprising:

establishing a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receiving a network message related to an electronic commerce transaction at the eTA system, ~~which is responsible for the communications between the network client and the network server;~~

identifying a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

generating a transaction identifier associated with ~~each electronic transaction~~ the received message received from the client and storing the transaction identifier information with the transaction type and message parameters at a back end database; and

preserving a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction; and

~~resuming~~ masking a failure of the electronic commerce transaction ~~from a failure by providing an expected response message to said received message~~ based upon the preserved state at the failure.

54. (Currently Amended) A system that determines the outcome of an electronic commerce transaction initiated by a network message between a network client and a network server, the system comprising one or more processors that execute program instructions and receive a data set, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the network server at an electronic transaction assurance (eTA) system;

receive a network message related to an electronic commerce transaction at the eTA system, ~~which is responsible for the communications between the network client and the network server;~~

identify a transaction type and message parameters included in the received network message, thereby defining an electronic commerce transaction to which the message relates;

generate a transaction identifier associated with ~~each electronic transaction~~ the received from the client message and storing the transaction identifier information with the transaction type and message parameters at a back end database; and

preserve a state of the electronic commerce transaction and updating the transaction type and message parameters in response to processing of the electronic commerce transaction; and

resume masking a failure of the electronic commerce transaction ~~from a failure~~ by providing an expected response message to said received message based upon the preserved state at the failure.

55. (Currently Amended) A program product for use in a processor that executes program steps recorded in a computer-readable media to perform a method for measuring the end-to-end response time of each electronic commerce transaction message sent from a network client side to a network server side of a Web site, the program product comprising:

a recordable media;

a program of computer-readable instructions executable by the processor to perform operations comprising:

establishing a communications connection between the network client and the Web site network server;

receiving a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message parameters, thereby defining an electronic commerce transaction to which the message relates;

adding code to the Web page served to the network client that records the time when a client request message is sent by the network client, indicating the start

of an electronic commerce transaction, and when a response message is received by the client, indicating the end of a said electronic commerce transaction;

generating a transaction identifier associated with each electronic commerce transaction request message received from the client and storing the transaction identifier information with the transaction type and message parameters at a back end database; and

preserving a state of the electronic transaction and updating the transaction type and message parameters in response to processing of the electronic transaction; and

resuming the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the request message from the network client.

56. (Currently Amended) A system that measures the end-to-end response time of each electronic commerce transaction message sent from a network client side to a network server side of a Web site, the system comprising one or more processors that execute program instructions and receive a data set, wherein the program instructions are executed to cause the processor to:

establish a communications connection between the network client and the Web site network server;

receive a network message from the network client, comprising a request for a Web site page such that the request identifies a transaction type and message

parameters, thereby defining an electronic commerce transaction to which the message relates;

add code to the Web page served to the network client that records the time when a client request message is sent by the network client, indicating the start of an electronic commerce transaction, and when a response is received by the client, indicating the end of a said electronic commerce transaction;

generate a transaction identifier associated with each ~~electronic transaction~~ received from the client and store the transaction identifier information with the transaction type and message parameters at a back end database; ~~and~~

preserve a state of the electronic transaction and update the transaction type and message parameters in response to processing of the electronic transaction; ~~and~~

resume the electronic transaction from a failure based upon the preserved state at the failure; and

masking the failure by providing an expected response to the request message from the network client.